# Extending Wagner's Hierarchy to Deterministic Visibly Pushdown Automata


Victor Selivanov [1, 2]

[1]Saint-Petersburg State University

[2]A.P. Ershov Institute of Informatics Systems, Novosibirsk



CiE-2023, Batumi (Georgia), July 25

# Introduction

K. Wagner [16] characterized the structure of regular $\omega$-languages under the continuous reducibility (known also as Wadge reducibility). Later, some results from [16] were extended to languages recognized by more complicated computing devices (see e.g. [4, 11, 5] and references therein for an extensive study of, in particular, context-free $\omega$-languages). In this wider context, some important properties of the Wagner hierarchy (e.g., the decidability of levels) usually fail.

# Introduction

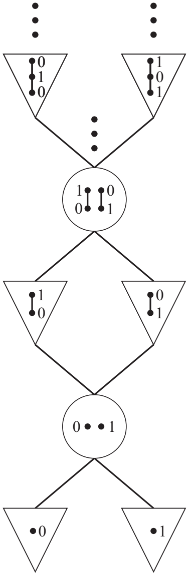
K. Wagner [16] characterized the structure of regular $\omega$-languages under the continuous reducibility (known also as Wadge reducibility). Later, some results from [16] were extended to languages recognized by more complicated computing devices (see e.g. [4, 11, 5] and references therein for an extensive study of, in particular, context-free $\omega$-languages). In this wider context, some important properties of the Wagner hierarchy (e.g., the decidability of levels) usually fail.

Motivated by a related study in descriptive set theory, computability and complexity, we extended in [13, 15] the Wagner theory from the regular sets to the regular $k$-partitions $A : \Sigma^\omega \to \{0, \ldots, k-1\} = \bar{k}$ of the set $\Sigma^\omega$ of $\omega$-words over a finite alphabet $\Sigma$ that essentially coincide with the $k$-tuples $(A_0, \ldots, A_{k-1})$ of pairwise disjoint regular sets satisfying $A_0 \cup \cdots \cup A_{k-1} = \Sigma^\omega$ (note that the $\omega$-languages are in a bijective correspondence with the 2-partitions of $\Sigma^\omega$).

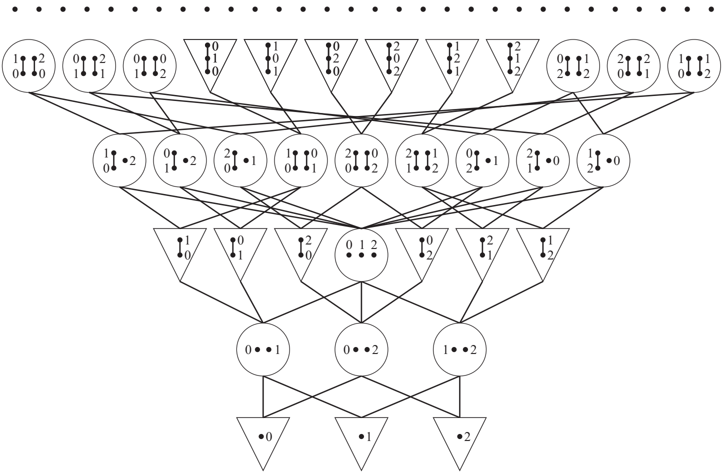
The extension from sets to $k$-partitions for $k > 2$ is non-trivial. It required to develop a machinery of iterated labeled trees and of the FH of $k$-partitions (partially systematized in [15]) which turned out crucial for the subsequent partial extension of the Wadge theory to $k$-partitions [14] and, as a concluding step, to $Q$-partitions for arbitrary better quasiorder $Q$ [6].

The extension from sets to $k$-partitions for $k > 2$ is non-trivial. It required to develop a machinery of iterated labeled trees and of the FH of $k$-partitions (partially systematized in [15]) which turned out crucial for the subsequent partial extension of the Wadge theory to $k$-partitions [14] and, as a concluding step, to $Q$-partitions for arbitrary better quasiorder $Q$ [6].

Among many natural extensions of regular $\omega$-languages, the class of visibly pushdown $\omega$-languages (also known as languages recognized by input-driven automata) [8, 3, 1, 2] and its subclasses seem especially interesting because they preserve many important properties of regular $\omega$-languages. An investigation of Wagner's hierarchy for this class was initiated in [10] where the hierarchy was successfully extended to the class of languages of well nested words.




An initial segment of $(\mathcal{F}_{\mathcal{T}_2}; \leq_h)$.



An initial segment of $(\mathcal{F}_{\mathcal{T}_3};\leq_h)$.

In this work we make the next step by extending the Wagner hierarchy in two directions: from the well nested words to the words of bounded stack height, and from languages to $k$-partitions. Our main result gives a complete effective characterization of the corresponding structures.

In this work we make the next step by extending the Wagner hierarchy in two directions: from the well nested words to the words of bounded stack height, and from languages to $k$-partitions. Our main result gives a complete effective characterization of the corresponding structures.

Although DVPA on words of bounded stack height have many common features with DFA, many details are different and require new notions and facts which are interesting on their own. The proof techniques here are quite different from those in [10]. The effective extensions of Wagner's theory may be of interest not only for automata theory of infinite words but also for the descriptive set theory because they help to identify the constructive part of (in general, very non-constructive) Wadge hierarchy.

In this work we make the next step by extending the Wagner hierarchy in two directions: from the well nested words to the words of bounded stack height, and from languages to $k$-partitions. Our main result gives a complete effective characterization of the corresponding structures.

Although DVPA on words of bounded stack height have many common features with DFA, many details are different and require new notions and facts which are interesting on their own. The proof techniques here are quite different from those in [10]. The effective extensions of Wagner's theory may be of interest not only for automata theory of infinite words but also for the descriptive set theory because they help to identify the constructive part of (in general, very non-constructive) Wadge hierarchy.

The general case of DVPA (and of non-deterministic VPA) will probably require new methods and techniques, because the set of Wadge degrees occupied by the corresponding languages and $k$-partitions is richer than the Wadge degrees of DVP languages of words of bounded stack height, as it follows from the results in [7] and from our work.

# Visibly pushdown automata

In the context of visibly pushdown automata, the input alphabet $\Sigma$ is split into three disjoint sets of *left brackets* $\Sigma_{+1}$, *right brackets* $\Sigma_{-1}$ and *neutral symbols* $\Sigma_0$. In this paper, symbols from $\Sigma_{+1}$ and $\Sigma_{-1}$ shall be denoted by left and right angled brackets, respectively ($<$, $>$), whereas lower-case Latin letters from the beginning of the alphabet ($a, b \ldots$) shall be used for symbols from $\Sigma_0$. Usually we work with a fixed such alphabet which is often omitted from the corresponding notation.

# Visibly pushdown automata

In the context of visibly pushdown automata, the input alphabet $\Sigma$ is split into three disjoint sets of *left brackets* $\Sigma_{+1}$, *right brackets* $\Sigma_{-1}$ and *neutral symbols* $\Sigma_0$. In this paper, symbols from $\Sigma_{+1}$ and $\Sigma_{-1}$ shall be denoted by left and right angled brackets, respectively ($<$, $>$), whereas lower-case Latin letters from the beginning of the alphabet ($a, b \ldots$) shall be used for symbols from $\Sigma_0$. Usually we work with a fixed such alphabet which is often omitted from the corresponding notation.

Along with the input alphabet $\Sigma$, a visibly pushdown automaton has a non-empty finite *stack alphabet* $\Gamma$ and a special symbol $\perp \notin \Gamma$ which denotes the bottom of the stack. The variable $\gamma$ is used below to denote elements of $\Gamma$, while variables $x, y, \ldots$ denote finite words over $\Gamma$ (which are called stack contents).

By a *deterministic visibly pushdown automaton (DVPA)* over $\Sigma$ we mean a tuple $\mathcal{M} = (Q, in, \Gamma, \{\delta^c\}_{c\in\Sigma})$ consisting of a finite non-empty set $Q$ of states, an initial state $in \in Q$, a stack alphabet $\Gamma$, and a family $\{\delta^c\}_{c\in\Sigma}$ of transition functions where $\delta^c : Q \to Q$ for $c \in \Sigma_0$, $\delta^c : Q \to Q \times \Gamma$ for $c \in \Sigma_{+1}$, and $\delta^c : Q \times (\Gamma \cup \{\perp\}) \to Q$ for $c \in \Sigma_{-1}$.

By a *deterministic visibly pushdown automaton (DVPA)* over $\Sigma$ we mean a tuple $\mathcal{M} = (Q, in, \Gamma, \{\delta^c\}_{c\in\Sigma})$ consisting of a finite non-empty set $Q$ of states, an initial state $in \in Q$, a stack alphabet $\Gamma$, and a family $\{\delta^c\}_{c\in\Sigma}$ of transition functions where $\delta^c : Q \to Q$ for $c \in \Sigma_0$, $\delta^c : Q \to Q \times \Gamma$ for $c \in \Sigma_{+1}$, and $\delta^c : Q \times (\Gamma \cup \{\bot\}) \to Q$ for $c \in \Sigma_{-1}$.

We call elements of $Q \times \Gamma^*$ *configurations of* $\mathcal{M}$ and denote them as $(q, x), (r, y), \ldots$. Configuration $(in, \varepsilon)$ is called the *initial configuration of* $\mathcal{M}$. Any letter $c \in \Sigma$ induces a unary function $(q, x) \mapsto (q, x) \cdot c$ on $Q \times \Gamma^*$ by the following rules: $(q, x) \cdot a = (\delta^a(q), x)$, $(q, x) \cdot <= (r, x \cdot \gamma)$ where $(r, \gamma) = \delta^<(q)$, $(q, \varepsilon) \cdot >= (\delta^>(q, \bot), \varepsilon)$, and $(q, x \cdot \gamma) \cdot >= (\delta^>(q, \gamma), x)$. Note that the top stack symbol in $(q, x)$ is the last symbol of $x$ (or $\bot$, if $x = \varepsilon$), i.e., the push and pop operations are performed on the right end of the stack content $x$. Subsets of $Q$ (i.e., elements of $P(Q)$) are sometimes called *macrostates of* $\mathcal{M}$.

Any finite word $u \in \Sigma^*$ induces a unary function $(q, x) \mapsto (q, x) \cdot u$ on $Q \times \Gamma^*$ by induction: $(q, x)\varepsilon = (q, x)$ and $(q, x) \cdot (u \cdot c) = ((q, x) \cdot u) \cdot c$. For $\alpha \in \Sigma^\omega$, by $(q, x) \cdot \alpha$ we denote the sequence of configurations $\{(q, x) \cdot \alpha \restriction_i\}_{i<\omega}$ called the $\alpha$-run of $\mathcal{M}$ starting with $(q, x)$. Abusing the notation $(q, x) \cdot u$ above, we also use it to denote the $u$-run of $\mathcal{M}$ starting with $(q, x)$, i.e., the sequence $\{(q, x) \cdot u \restriction_i\}_{i<|u|}$.

Any finite word $u \in \Sigma^*$ induces a unary function $(q, x) \mapsto (q, x) \cdot u$ on $Q \times \Gamma^*$ by induction: $(q, x)\varepsilon = (q, x)$ and $(q, x) \cdot (u \cdot c) = ((q, x) \cdot u) \cdot c$. For $\alpha \in \Sigma^\omega$, by $(q, x) \cdot \alpha$ we denote the sequence of configurations $\{(q, x) \cdot \alpha \restriction_i\}_{i<\omega}$ called the $\alpha$-run of $\mathcal{M}$ starting with $(q, x)$. Abusing the notation $(q, x) \cdot u$ above, we also use it to denote the $u$-run of $\mathcal{M}$ starting with $(q, x)$, i.e., the sequence $\{(q, x) \cdot u \restriction_i\}_{i<|u|}$.

Define also the function $f_{\mathcal{M}} : \Sigma^\omega \to P(Q)$ by setting $f_{\mathcal{M}}(\alpha)$ to be the set of states which occur infinitely often in the $\alpha$-run $(in, \varepsilon)\alpha$. We say that a configuration $(r, y)$ is *reachable from* $(q, x)$ if $(r, y) = (q, x)u$ for some $u \in \Sigma^*$. The configuration $(r, y)$ is *reachable* if it is reachable from the initial state $(in, \varepsilon)$.

Any finite word $u \in \Sigma^*$ induces a unary function $(q, x) \mapsto (q, x) \cdot u$ on $Q \times \Gamma^*$ by induction: $(q, x)\varepsilon = (q, x)$ and $(q, x) \cdot (u \cdot c) = ((q, x) \cdot u) \cdot c$. For $\alpha \in \Sigma^\omega$, by $(q, x) \cdot \alpha$ we denote the sequence of configurations $\{(q, x) \cdot \alpha \restriction_i\}_{i<\omega}$ called the $\alpha$-run of $\mathcal{M}$ starting with $(q, x)$. Abusing the notation $(q, x) \cdot u$ above, we also use it to denote the $u$-run of $\mathcal{M}$ starting with $(q, x)$, i.e., the sequence $\{(q, x) \cdot u \restriction_i\}_{i<|u|}$.

Define also the function $f_{\mathcal{M}} : \Sigma^\omega \to P(Q)$ by setting $f_{\mathcal{M}}(\alpha)$ to be the set of states which occur infinitely often in the $\alpha$-run $(in, \varepsilon)\alpha$. We say that a configuration $(r, y)$ is *reachable from* $(q, x)$ if $(r, y) = (q, x)u$ for some $u \in \Sigma^*$. The configuration $(r, y)$ is *reachable* if it is reachable from the initial state $(in, \varepsilon)$.

Similar to DFA, for DVPA there are natural operations of Cartesian product. We briefly define one of these. Given DVPA $\mathcal{M}_1 = (Q_1, in_1, \Gamma_1, \{\delta_1^c\}_{c\in\Sigma})$ and $\mathcal{M}_2 = (Q_2, in_2, \Gamma_2, \{\delta_2^c\}_{c\in\Sigma})$ with stack bottom letters $\perp_1, \perp_2$, their product $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2 = (Q, in, \Gamma, \{\delta^c\}_{c\in\Sigma})$, with stack bottom letter $\perp = (\perp_1, \perp_2)$, is defined by:

$Q = Q_1 \times Q_2$, $in = (in_1, in_2)$, $\Gamma = \Gamma_1 \times \Gamma_2$,
$\delta^a(q_1, q_2) = (\delta_1^a(q_1), \delta_2^a(q_2))$,
$\delta^<(q_1, q_2) = ((r_1, r_2), (\gamma_1, \gamma_2))$ where $\delta_1^<(q_1) = (r_1, \gamma_1)$ and
$\delta_2^<(q_2) = (r_2, \gamma_2)$,
$\delta^>((q_1, q_2), \perp) = (\delta_1^>(q_1, \perp_1), \delta_2^>(q_2, \perp_2))$, and
$\delta^>((q_1, q_2), (\gamma_1, \gamma_2)) = (\delta_1^a(q_1, \gamma_1), \delta_2^a(q_2, \gamma_2))$.

$Q = Q_1 \times Q_2$, $in = (in_1, in_2)$, $\Gamma = \Gamma_1 \times \Gamma_2$,
$\delta^a(q_1, q_2) = (\delta_1^a(q_1), \delta_2^a(q_2))$,
$\delta^<(q_1, q_2) = ((r_1, r_2), (\gamma_1, \gamma_2))$ where $\delta_1^<(q_1) = (r_1, \gamma_1)$ and
$\delta_2^<(q_2) = (r_2, \gamma_2)$,
$\delta^>((q_1, q_2), \perp) = (\delta_1^>(q_1, \perp_1), \delta_2^>(q_2, \perp_2))$, and
$\delta^>((q_1, q_2), (\gamma_1, \gamma_2)) = (\delta_1^a(q_1, \gamma_1), \delta_2^a(q_2, \gamma_2))$.

Note that the behaviour of $\mathcal{M}$ "includes" the behaviours of $\mathcal{M}_1$ and $\mathcal{M}_2$, in particular $f_{\mathcal{M}}(\alpha) = f_{\mathcal{M}_1}(\alpha) \times f_{\mathcal{M}_2}(\alpha)$.

$Q = Q_1 \times Q_2$, $in = (in_1, in_2)$, $\Gamma = \Gamma_1 \times \Gamma_2$,
$\delta^a(q_1, q_2) = (\delta_1^a(q_1), \delta_2^a(q_2))$,
$\delta^<(q_1, q_2) = ((r_1, r_2), (\gamma_1, \gamma_2))$ where $\delta_1^<(q_1) = (r_1, \gamma_1)$ and $\delta_2^<(q_2) = (r_2, \gamma_2)$,
$\delta^>((q_1, q_2), \perp) = (\delta_1^>(q_1, \perp_1), \delta_2^>(q_2, \perp_2))$, and
$\delta^>((q_1, q_2), (\gamma_1, \gamma_2)) = (\delta_1^a(q_1, \gamma_1), \delta_2^a(q_2, \gamma_2))$.

Note that the behaviour of $\mathcal{M}$ "includes" the behaviours of $\mathcal{M}_1$ and $\mathcal{M}_2$, in particular $f_{\mathcal{M}}(\alpha) = f_{\mathcal{M}_1}(\alpha) \times f_{\mathcal{M}_2}(\alpha)$.

A *DVP Muller acceptor* is a pair $(\mathcal{M}, A)$ where $\mathcal{M}$ is a DVPA and $A \subseteq P(Q)$ is a set of macrostates; it recognizes the set $L(\mathcal{M}, A) = f_{\mathcal{M}}^{-1}(A)$. The $\omega$-languages recognized by such acceptors are called DVP-sets. The class $\mathcal{D}$ of DVP-sets is closed under the Boolean operations [1] and is contained in the Boolean closure $BC(\mathbf{\Sigma}_2^0)$ of the second level $\mathbf{\Sigma}_2^0$ of Borel hierarchy in $\Sigma^\omega$.

A *DVP $k$-partition* is a $k$-partition $A : \Sigma^\omega \to \bar{k}$ all of whose components are DVP-sets. A DVP $k$-partition $A$ may be specified by a $k$-tuple of DVP Muller acceptors which recognize the components $A_0, \ldots, A_{k-1}$ but for our purposes we need a slightly different presentation similar to that used in [13, 15]. A *DVP Muller $k$-acceptor* is a pair $(\mathcal{M}, A)$ where $\mathcal{M}$ is a DVPA and $A : P(Q) \to \bar{k}$ is a $k$-partition of $P(Q)$. The DVP Muller $k$-acceptor recognises the DVP $k$-partition $L(\mathcal{M}, A) = A \circ f_{\mathcal{M}}$ where $f_{\mathcal{M}} : \Sigma^\omega \to P(Q)$ is defined above.

A *DVP $k$-partition* is a $k$-partition $A : \Sigma^\omega \to \bar{k}$ all of whose components are DVP-sets. A DVP $k$-partition $A$ may be specified by a $k$-tuple of DVP Muller acceptors which recognize the components $A_0, \ldots, A_{k-1}$ but for our purposes we need a slightly different presentation similar to that used in [13, 15]. A *DVP Muller $k$-acceptor* is a pair $(\mathcal{M}, A)$ where $\mathcal{M}$ is a DVPA and $A : P(Q) \to \bar{k}$ is a $k$-partition of $P(Q)$. The DVP Muller $k$-acceptor recognises the DVP $k$-partition $L(\mathcal{M}, A) = A \circ f_{\mathcal{M}}$ where $f_{\mathcal{M}} : \Sigma^\omega \to P(Q)$ is defined above.

## Proposition

*A $k$-partition $L : X^\omega \to \bar{k}$ is DVP iff it is recognised by a DVP Muller $k$-acceptor.*

A *DVP $k$-partition* is a $k$-partition $A : \Sigma^\omega \to \bar{k}$ all of whose components are DVP-sets. A DVP $k$-partition $A$ may be specified by a $k$-tuple of DVP Muller acceptors which recognize the components $A_0, \ldots, A_{k-1}$ but for our purposes we need a slightly different presentation similar to that used in [13, 15]. A *DVP Muller $k$-acceptor* is a pair $(\mathcal{M}, A)$ where $\mathcal{M}$ is a DVPA and $A : P(Q) \to \bar{k}$ is a $k$-partition of $P(Q)$. The DVP Muller $k$-acceptor recognises the DVP $k$-partition $L(\mathcal{M}, A) = A \circ f_{\mathcal{M}}$ where $f_{\mathcal{M}} : \Sigma^\omega \to P(Q)$ is defined above.

## Proposition

*A $k$-partition $L : X^\omega \to \bar{k}$ is DVP iff it is recognised by a DVP Muller $k$-acceptor.*

## Proposition

*For every $\Sigma$ we have: $\mathcal{R}_\Sigma \subseteq \mathcal{D}_\Sigma$. If at least one of $\Sigma_{+1}, \Sigma_{-1}$ is empty then $\mathcal{R}_\Sigma = \mathcal{D}_\Sigma$, otherwise the inclusion $\mathcal{R}_\Sigma \subset \mathcal{D}_\Sigma$ is proper, even for languages of well-nested words. Similarly for the regular and DVP $k$-partitions.*

# Stack height factorizations

Define the *stack height function* $sh : \Sigma^* \to \omega$ by induction: $sh(\varepsilon) = 0$, $sh(u \cdot a) = sh(u)$, $sh(u \cdot <) = sh(u) + 1$, $sh(u \cdot >) = min\{sh(u) - 1, 0\}$. Let $L_{\mathrm{mwm}}$ be the set of *minimally well-matched words*, i.e., words of the form $< w >$ where the last letter $>$ matches the first letter $<$, and $sh(< v)$ is positive for every $v \sqsubseteq w$. Then $W = (\Sigma_0 \cup L_{\mathrm{mwm}})^*$ is the set of *well-matched* (also known as *well-nested*) words. We also define the sets $W_{-1} = (\Sigma_{-1} \cup \Sigma_0 \cup L_{\mathrm{mwm}})^*$ of words without unmatched left brackets, and $W_{+1} = (\Sigma_{+1} \cup \Sigma_0 \cup L_{\mathrm{mwm}})^*$ of words without unmatched right brackets. Then $W = W_{-1} \cap W_{+1}$ and $W_{-1} = \{u \in \Sigma^* \mid sh(u) = 0\}$.Note that $\Sigma^* = W_{-1} \cdot W_{+1}$, and $(W_{-1}, W_{-1} \cdot (W_{+1} \setminus W))$ is a partition of $\Sigma^*$.

# Stack height factorizations

Define the *stack height function* $sh : \Sigma^* \to \omega$ by induction: $sh(\varepsilon) = 0$, $sh(u \cdot a) = sh(u)$, $sh(u \cdot <) = sh(u) + 1$, $sh(u \cdot >) = min\{sh(u) - 1, 0\}$. Let $L_{\mathrm{mwm}}$ be the set of *minimally well-matched words*, i.e., words of the form $< w >$ where the last letter $>$ matches the first letter $<$, and $sh(< v)$ is positive for every $v \sqsubseteq w$. Then $W = (\Sigma_0 \cup L_{\mathrm{mwm}})^*$ is the set of *well-matched* (also known as *well-nested*) words. We also define the sets $W_{-1} = (\Sigma_{-1} \cup \Sigma_0 \cup L_{\mathrm{mwm}})^*$ of words without unmatched left brackets, and $W_{+1} = (\Sigma_{+1} \cup \Sigma_0 \cup L_{\mathrm{mwm}})^*$ of words without unmatched right brackets. Then $W = W_{-1} \cap W_{+1}$ and $W_{-1} = \{u \in \Sigma^* \mid sh(u) = 0\}$.Note that $\Sigma^* = W_{-1} \cdot W_{+1}$, and $(W_{-1}, W_{-1} \cdot (W_{+1} \setminus W))$ is a partition of $\Sigma^*$.

For any $n < \omega$, let $H_n = \{w \in \Sigma^+ \mid sh(w) = n\}$ be the set of non-empty words of stack height $n$. Then $\{H_n\}_n$ is a partition of $\Sigma^+$, and $H_n$ coincides with the set of non-empty words which have precisely $n$ unmatched left brackets.

## Proposition

$H_0 = W_{-1} \setminus \{\varepsilon\}$, $H_1 = W_{-1} \cdot < \cdot W$, *and* $H_{n+2} = H_{n+1} \cdot < \cdot W$.

## Proposition

*$H_0 = W_{-1} \setminus \{\varepsilon\}$, $H_1 = W_{-1} \cdot < \cdot W$, and $H_{n+2} = H_{n+1} \cdot < \cdot W$.*

For any $n \leq \omega$, let $G_n$ be the set of infinite words which have precisely $n$ unmatched left brackets, then $\{G_n\}_{n \leq \omega}$ is a partition of $\Sigma^\omega$. Let $\mathrm{wn} = \mathrm{wn}(\Sigma) = (\Sigma_0 \cup L_{\mathrm{mwm}})^\omega$ be the set of well-matched infinite words over $\Sigma$. Let also $G_{<\omega} = \bigcup_{n<\omega} G_n$ and $G_{\leq n} = G_0 \cup \cdots \cup G_n$ for $n < \omega$.

## Proposition

*$G_0 = (\Sigma_{-1} \cup \Sigma_0 \cup L_{\mathrm{mwm}})^\omega$, $G_1 = W_{-1} \cdot < \cdot \mathrm{wn}$, $G_{n+2} = H_{n+1} \cdot < \cdot \mathrm{wn}$ for $n < \omega$, and $G_\omega = W_{-1} \cdot (< \cdot W)^\omega$.*

For any $w \in \Sigma^+$ (resp. $\alpha \in \Sigma^\omega$), we can group maximal subwords which are in $L_{\mathrm{mwm}}$, and obtain a unique factorization (which we call sh-factorization) $w = w_0 \cdots w_n$ (resp. $\alpha = w_0 w_1 \cdots$) where each factor $w_i$ is in $\Sigma \cup L_{\mathrm{mwm}}$.

For any $w \in \Sigma^+$ (resp. $\alpha \in \Sigma^\omega$), we can group maximal subwords which are in $L_{\mathrm{mwm}}$, and obtain a unique factorization (which we call sh-factorization) $w = w_0 \cdots w_n$ (resp. $\alpha = w_0 w_1 \cdots$) where each factor $w_i$ is in $\Sigma \cup L_{\mathrm{mwm}}$.

In the case of an infinite word $\alpha$, the sh-factorization is also determined by the set $S_\alpha = \{n < \omega \mid \forall m \geq n(sh(\alpha \restriction_m) \geq sh(\alpha \restriction_n))\}$ (see Section 3 of [7] where $S_\alpha$ is denoted as $Steps_\alpha$), namely $w_i = \alpha[n_i, n_{i+1})$ for every $i < \omega$ where $S_\alpha = \{0 = n_0 < n_1 < \cdots\}$. A similar description exists to the case of a finite word $w$, only now the set $S_w$ is finite.

For any $w \in \Sigma^+$ (resp. $\alpha \in \Sigma^\omega$), we can group maximal subwords which are in $L_{\mathrm{mwm}}$, and obtain a unique factorization (which we call sh-factorization) $w = w_0 \cdots w_n$ (resp. $\alpha = w_0 w_1 \cdots$) where each factor $w_i$ is in $\Sigma \cup L_{\mathrm{mwm}}$.

In the case of an infinite word $\alpha$, the sh-factorization is also determined by the set $S_\alpha = \{n < \omega \mid \forall m \geq n(sh(\alpha \restriction_m) \geq sh(\alpha \restriction_n))\}$ (see Section 3 of [7] where $S_\alpha$ is denoted as $Steps_\alpha$), namely $w_i = \alpha[n_i, n_{i+1})$ for every $i < \omega$ where $S_\alpha = \{0 = n_0 < n_1 < \cdots\}$. A similar description exists to the case of a finite word $w$, only now the set $S_w$ is finite.

## Proposition

1. *The sets $L_{\mathrm{mwm}}, W, W_{-1}, W_{+1}$ are computable, and $H_n$ is computable uniformly on $n$.*
2. *The sh-factorization function on $\Sigma^+$ is computable.*

# Cycles of DVPA

A basic notion in the study of Wagner hierarchy is the notion of a cycle of a DFA (see [16, 15]). The same applies to DVPA but in this case cycles are a bit more complex.

# Cycles of DVPA

A basic notion in the study of Wagner hierarchy is the notion of a cycle of a DFA (see [16, 15]). The same applies to DVPA but in this case cycles are a bit more complex.

## Definition

*A cycle of a DVPA $\mathcal{M}$ is a pair $\mathrm{c} = (u, (q, x))$ where $u$ is a nonempty word from $W_{-1} \cup W_{+1}$, $(q, x)$ is a reachable configuration of $\mathcal{M}$, $x = \varepsilon$ whenever $u \in W_{-1} \setminus W$, and $(q, x)u = (q, y)$ for some $y \in \Gamma^*$. In the case $u \in W_{-1}$, we call $sh(\mathrm{c}) = |x|$ the stack height of $\mathrm{c}$.*

# Cycles of DVPA

A basic notion in the study of Wagner hierarchy is the notion of a cycle of a DFA (see [16, 15]). The same applies to DVPA but in this case cycles are a bit more complex.

## Definition

*A cycle of a DVPA $\mathcal{M}$ is a pair $\mathrm{c} = (u, (q, x))$ where $u$ is a nonempty word from $W_{-1} \cup W_{+1}$, $(q, x)$ is a reachable configuration of $\mathcal{M}$, $x = \varepsilon$ whenever $u \in W_{-1} \setminus W$, and $(q, x)u = (q, y)$ for some $y \in \Gamma^*$. In the case $u \in W_{-1}$, we call $sh(\mathrm{c}) = |x|$ the stack height of $\mathrm{c}$.*

The set of cycles of $\mathcal{M}$ is denoted by $C_{\mathcal{M}}$. The set of states in the run $(q, x)u$ is called the macrostate of $\mathrm{c}$ and denoted as $M(\mathrm{c})$, so $M : C_{\mathcal{M}} \to P(Q)$. Let $C_{\mathcal{M}}^{\leq n} = \{\mathrm{c} \in C_{\mathcal{M}} \mid u \in W_{-1} \wedge sh(\mathrm{c}) \leq n\}$ for every $n < \omega$. We say that a cycle $\mathrm{d} = (v, (r, y)) \in C_{\mathcal{M}}$ is reachable from $\mathrm{c}$ if $(r, y)$ is reachable from $(q, x)$.

For any $u \in \Sigma^*$ and $q \in Q$, let $\tilde{u}_q$ denote the unique stack content such that $(q, \varepsilon)u = (r, \tilde{u}_q)$ for some $r \in Q$. Note that $\tilde{u}_q = \varepsilon$ whenever $u \in W_{-1}$. For any $\mathrm{c} = (u, (q, x)) \in C_{\mathcal{M}}$ and $n \geq 1$, let $\mathrm{c}^n = (u^n, (q, x))$.

## Lemma

*Let* $\mathrm{c} = (u, (q, x)) \in C_{\mathcal{M}}$ *and* $n \geq 1$. *Then* $(q, x)u^n = (u, (q, x \cdot \tilde{u}_q^n))$, *the pairs* $\mathrm{c}^n$, *and* $(u, (q, x \cdot \tilde{u}_q^n))$ *are cycles of* $\mathcal{M}$, *and* $M(\mathrm{c}^n) = M(\mathrm{c}) = M((u, x \cdot \tilde{u}_q^n))$.

For any $u \in \Sigma^*$ and $q \in Q$, let $\tilde{u}_q$ denote the unique stack content such that $(q, \varepsilon)u = (r, \tilde{u}_q)$ for some $r \in Q$. Note that $\tilde{u}_q = \varepsilon$ whenever $u \in W_{-1}$. For any $\mathrm{c} = (u, (q, x)) \in C_{\mathcal{M}}$ and $n \geq 1$, let $\mathrm{c}^n = (u^n, (q, x))$.

### Lemma

*Let $\mathrm{c} = (u, (q, x)) \in C_{\mathcal{M}}$ and $n \geq 1$. Then $(q, x)u^n = (u, (q, x \cdot \tilde{u}_q^n))$, the pairs $\mathrm{c}^n$, and $(u, (q, x \cdot \tilde{u}_q^n))$ are cycles of $\mathcal{M}$, and $M(\mathrm{c}^n) = M(\mathrm{c}) = M((u, x \cdot \tilde{u}_q^n))$.*

We define a function $g_{\mathcal{M}} : C_{\mathcal{M}} \to \Sigma^\omega$ by setting $g_{\mathcal{M}}(\mathrm{c}) = u' \cdot u^\omega$ where $\mathrm{c} = (u, (q, x))$ and $u'$ is a word with $(in, \varepsilon)u' = (q, x)$.

### Lemma

*For any $\mathrm{c} = (u, (q, x)) \in C_{\mathcal{M}}$ we have $f_{\mathcal{M}}(g_{\mathcal{M}}(\mathrm{c})) = M(\mathrm{c})$. If $u \in W_{-1} \wedge x = \varepsilon$ (resp. $u \in W \wedge x \neq \varepsilon$, $u \in W_{+1} \setminus W$) then $g_{\mathcal{M}}(\mathrm{c})$ is in $G_0$ (resp. $G_{n+1}$ for some $n < \omega$, $G_\omega$).*

### Lemma

*For any $\alpha \in \Sigma^\omega$ there is $\mathrm{c} = \mathrm{c}_{\mathcal{M}}(\alpha) \in C_{\mathcal{M}}$ such that $M(\mathrm{c}) = f_{\mathcal{M}}(\alpha)$. Furthermore, $\mathrm{c}_{\mathcal{M}}(\alpha) \in C_{\mathcal{M}}^{\leq n}$ for $\alpha \in G_{\leq n}$, and $\{M(\mathrm{c}) \mid \mathrm{c} \in C_{\mathcal{M}}\} = \{f_{\mathcal{M}}(\alpha) \mid \alpha \in \Sigma^\omega\}$.*

# DVP $\omega$-languages in Borel hierarchy

## Definition

*Let* $\mathrm{c} = (u, (q, x))$ *and* $\mathrm{d} = (v, (r, y))$ *be in* $C_{\mathcal{M}}$*. Then* $\mathrm{c} \leq_0 \mathrm{d}$*, if for every* $m \geq 1$ *there is* $n \geq 1$ *such that* $\mathrm{d}^n$ *is reachable from* $\mathrm{c}^m$ *(i.e.,* $(r, y\tilde{v}_r^n)$ *is reachable from* $(q, x\tilde{u}_r^m)$*). Let also* $\mathrm{c} \leq_1 \mathrm{d}$ *mean that* $\mathrm{c} \equiv_0 \mathrm{d}$ *and* $M(\mathrm{c}) \supseteq M(\mathrm{d})$*.*

# DVP $\omega$-languages in Borel hierarchy

## Definition

*Let* $\mathrm{c} = (u, (q, x))$ *and* $\mathrm{d} = (v, (r, y))$ *be in* $C_{\mathcal{M}}$. *Then* $\mathrm{c} \leq_0 \mathrm{d}$, *if for every* $m \geq 1$ *there is* $n \geq 1$ *such that* $\mathrm{d}^n$ *is reachable from* $\mathrm{c}^m$ *(i.e.,* $(r, y\tilde{v}_r^n)$ *is reachable from* $(q, x\tilde{u}_r^m)$*). Let also* $\mathrm{c} \leq_1 \mathrm{d}$ *mean that* $\mathrm{c} \equiv_0 \mathrm{d}$ *and* $M(\mathrm{c}) \supseteq M(\mathrm{d})$*.*

Let $\mathcal{C}_i$ be the class of all $\leq_i$-up subsets of $C_{\mathcal{M}}^{\leq n}$ (a set $A \subseteq C_{\mathcal{M}}^{\leq n}$ is $\leq_i$-up if $\mathrm{a} \in A$ and $\mathrm{a} \leq_i \mathrm{c}$ imply $\mathrm{c} \in A$; $\leq_i$-down sets are defined similarly). The pair $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1)$ is called *the 2-base of up-sets over* $(C_{\mathcal{M}}^{\leq n}; \leq_0, \leq_1)$. According to [15], the 2-base $\mathcal{C}$ is *interpolable* if for every disjoint $\leq_1$-down sets $A, B \subseteq C_{\mathcal{M}}^{\leq n}$ there is a finite Boolean combination $C$ of $\mathcal{C}_0$-sets which separates $A$ from $B$, i.e., $A \subseteq C$ and $C \cap B = \emptyset$. The structure $(C_{\mathcal{M}}^{\leq n}; \leq_0, \leq_1)$ is a *2-preorder* if both $\leq_0, \leq_1$ are preorders, and $\mathrm{c} \leq_1 \mathrm{d}$ implies $\mathrm{c} \equiv_0 \mathrm{d}$; the 2-preorder is *compatible* if $\mathrm{c} \equiv_0 \mathrm{d}$ implies $\exists \mathrm{e}(\mathrm{e} \leq_1 \mathrm{c} \wedge \mathrm{e} \leq_1 \mathrm{d})$.

### Lemma

*For any DVPA $\mathcal{M}$ and any $n < \omega$, the structure $(C_{\mathcal{M}}^{\leq n}; \leq_0, \leq_1)$ is a compatible 2-preorder. This structure is computably presentable uniformly on $n$. There are only finitely many classes under the equivalence relation $\equiv_0$ induced by $\leq_0$. The 2-base $\mathcal{C}$ is interpolable.*

## Lemma

*For any DVPA $\mathcal{M}$ and any $n < \omega$, the structure $(C_{\mathcal{M}}^{\leq n}; \leq_0, \leq_1)$ is a compatible 2-preorder. This structure is computably presentable uniformly on $n$. There are only finitely many classes under the equivalence relation $\equiv_0$ induced by $\leq_0$. The 2-base $\mathcal{C}$ is interpolable.*

## Lemma

*Let $(\mathcal{M}, A)$ be a DVP Muller acceptor.*

1. *If the set $\mathcal{A} = \{\mathrm{c} \in C_{\mathcal{M}}^{\leq n} \mid M(\mathrm{c}) \in A\}$ is $\leq_0$-up then the set $L_A = \{\alpha \in G_{\leq n} \mid f_{\mathcal{M}}(\alpha) \in A\}$ is in $\mathbf{\Sigma}_1(G_{\leq n})$, otherwise $L_A$ is $\mathbf{\Pi}_1(G_{\leq n})$-hard w.r.t. the Wadge reducibility in $G_{\leq n}$.*
2. *If $\mathcal{A}$ is $\leq_1$-up then $L_A$ is in $\mathbf{\Sigma}_2(G_{\leq n})$, otherwise $L_A$ is $\mathbf{\Pi}_2(G_{\leq n})$-hard w.r.t. the Wadge reducibility in $G_{\leq n}$.*

## Main result

The Wadge reducibility for $k$-partitions $K, L : G_{\leq n} \to \bar{k}$ is defined as follows: $K \leq_W L$, if $K = L \circ g$ for some continuous function $g$ on $G_{\leq n}$. Let $\mathrm{DVP}^{(k)}_{\leq n}$ be the set of such $k$-partitions recognized by the DVP Muller $k$-acceptors. Let $\mathcal{T}_k$ be the set of finite $\bar{k}$-labeled trees with the homomorphic preorder. Let $(\mathcal{F}_{\mathcal{T}_k}; \leq_h)$ be the set of finite $\mathcal{T}_k$-labeled forests with the homomorphic preorder.

### Theorem

*For any $n < \omega$, the quotient posets of $(\mathrm{DVP}^{(k)}_{\leq n}; \leq_W)$ and of $(\mathcal{F}_{\mathcal{T}_k}; \leq_h)$ are computably isomorphic and computably presentable.*

# Main result

The Wadge reducibility for $k$-partitions $K, L : G_{\leq n} \to \bar{k}$ is defined as follows: $K \leq_W L$, if $K = L \circ g$ for some continuous function $g$ on $G_{\leq n}$. Let $\mathrm{DVP}^{(k)}_{\leq n}$ be the set of such $k$-partitions recognized by the DVP Muller $k$-acceptors. Let $\mathcal{T}_k$ be the set of finite $\bar{k}$-labeled trees with the homomorphic preorder. Let $(\mathcal{F}_{\mathcal{T}_k}; \leq_h)$ be the set of finite $\mathcal{T}_k$-labeled forests with the homomorphic preorder.

## Theorem

*For any $n < \omega$, the quotient posets of $(\mathrm{DVP}^{(k)}_{\leq n}; \leq_W)$ and of $(\mathcal{F}_{\mathcal{T}_k}; \leq_h)$ are computably isomorphic and computably presentable.*

## Corollary

*The quotient posets of $(\mathrm{DVP}^{(k)}_{\leq n}; \leq_W)$ and of $(\mathcal{R}_k; \leq_W)$ are isomorphic.*

Remark. The quotient posets of $(\mathrm{DVP}^{(k)}_{<\omega}; \leq_W)$ and of $(\mathcal{R}_k; \leq_W)$ are not isomorphic.

Thank you for your attention!

# References


[1] Alur R., Madhusudan P.: Visibly pushdown languages, *ACM Symposium on Theory of Computing* (STOC 2004, Chicago, USA, 13–16 June 2004), 202–211.

[2] Alur R., Madhusudan P.: Adding nesting structure to words, *Journal of the ACM*, 56:3 (2009).

[3] von Braunmühl B., Verbeek R.: Input driven languages are recognized in $\log n$ space, *Annals of Discrete Mathematics*, 24 (1985), 1–20.

[4] Duparc J.: A hierarchy of deterministic context-free $\omega$-languages, *Theoretical Computer Science*, 290(3) (2003) 1253–1300.

[5] Finkel O.: Topological complexity of context-free $\omega$-languages: a survey, In: Language, Culture, Computation: Studies in Honor of Yaacov Choueka. Ed. by Nachum Dershowitz and Ephraim Nissan, Part I, Computing, Theory and Technology, LNCS 8001, Springer, 2014.

[6] Kihara T., Montalbán A.: On the structure of the Wadge degrees of bqo-valued Borel functions, Trans. Amer. Math. Soc. 371 (2019), no. 11, 7885–7923.

[7] Löding C., Madhusudan P., Serre O.: Visibly pushdown games, FSTTCS 2004, 408–420.

[8] Mehlhorn K.: Pebbling mountain ranges and its application to DCFL-recognition, *Automata, Languages and Programming* (ICALP 1980, Noordweijkerhout, The Netherlands, 14–18 July 1980), LNCS 85, 422-435.

[9] Okhotin A., Salomaa K.: Complexity of input-driven pushdown automata, *SIGACT News*, 45:2 (2014), 47–67.

[10] Okhotin A., Selivanov V.: Input-driven automata on well-nested infinite strings: automata-theoretic and topological properties. R. Santhanam and D. Musatov (Eds.): CSR 2021, Lecture Notes in Computer Science, v. 12730, pp. 349–360, 2021.

[11] Selivanov V.L.: Wadge degrees of $\omega$-languages of deterministic Turing machines. *Theoretical Informatics and Applications*, 37 (2003), 67–83.

[12] Selivanov V.L.: Hierarchies and reducibilities on regular languages related to modulo counting. *RAIRO Theoretical Informatics and Applications*, 41 (2009), 95–132.

[13] Selivanov V.L.: A fine hierarchy of $\omega$-regular $k$-partitions. B. Löwe et.al. (Eds.): CiE 2011, LNCS v. 6735, pp. 260–269. Springer, Heidelberg (2011).

[14] Selivanov V.L.: Extending Wadge theory to k-partitions. J. Kari, F. Manea and Ion Petre (eds.) LNCS 10307, 2017, 387–399, Berlin, Springer.

[15] Selivanov V.: Wadge degrees of classes of $\omega$-regular $k$-partitions. To appear in Journal of Automata, Languages and Combinatorics. An earlier version: Arxiv 2104.10358.

[16] Wagner K.: On $\omega$-regular sets. *Information and Control*, 43 (1979), 123–177.